УДК 004

НЕЧЕТКАЯ МОДЕЛЬ АНАЛИЗА РИСКОВ УЯЗВИМОСТИ НА ОСНОВЕ ОТКРЫТЫХ ИСТОЧНИКОВ

Ю. В. Татаринова[1,2,a], О. И. Синельникова[2]

[1] *Харьковский национальный университет радиоэлектроники*
[2] *ТОВ. Самсунг Электроникс Украина Компани*

Аннотация

В статье описывается подход оценки воздействия уязвимости на программный продукт на основе нечеткой логике. Входные лингвистические переменные определяются на основании информации об уязвимости, полученной из открытых источников. Функции принадлежности, их параметры, а также набор продукционных правил определяются на основании экспертных данных. Выходная переменная характеризует степень риска уязвимости и приоритет в очереди для ее устранения. Предложенный подход позволяет провести анализ рисков с учетом неполного набора данных. Разработанная модель позволяет существенно снизить время обработки и анализа опубликованных уязвимостей.

Ключевые слова: CVE, система нейро-нечеткого вывода, анализ рисков, информационная безопасности

1. Введение

С ростом популярности программного продукта или устройства (смартфон, носимые гаджеты, IoT устройства) растет и число найденных уязвимостей, угроз и векторов атаки. В связи с этим возникает потребность в эффективной системе оценки рисков. Существующие системы (такие как CORAS [1], ISO 27001 [2], OCTAVE [3]) порой неспособны качественно и точно оценить возникшие риски или неприменимы для конкретной уязвимости или системы, и таким образом, не только затрудняют принятие соответствующего решения, но и задерживают сроки его принятия.

Использование метода на основе нечеткой логике в данном случае является наиболее приемлемым решением задачи анализа и приоритизации рисков уязвимости, поскольку очень часто компоненты анализа не упорядочены, не формализованы, а некоторые перемененные могут вообще отсутствовать или информация о них имеет двусмысленный характер.

Цель данной статьи создать модель оценки рисков ИБ, которая основана на анализе уязвимостей из открытых источников с применением нечеткой логики. Сформировать базу правил на основе экспертных данных. Эффективность предложенного подхода доказана экспертной оценкой и проверена на наборе уязвимостей из базы данных «Common Vulnerabilities and Exposures» (CVE) [4] .

2. Нечеткая модель для анализа рисков уязвимости

В устройствах, которые используют программное обеспечение (ПО) сторонних производителей (проприетарное ПО или проекты с открытым исходным кодом), находится большое количество уязвимостей, которые требуют анализа и приоритизации порядка их исправлений. Анализ и оценка каждой обнародованной уязвимости, которая попадает в открытый доступ является трудоемкой задачей как по человеческим ресурсам, так и временным интервалом. Второй проблемой является процесс исправления ошибок – не всегда есть ресурсы своевременно исправлять уязвимый код. Возникает задача в определении наиболее опасного набора уязвимостей на системе с учетом наличия информации о них в открытом доступе. Ведь именно эту информацию в первую очередь может использовать злоумышленник. Общая схема работы технологии, задача которой решить вышеперечисленные проблемы представлена в [5].

Базовые положения теории нечеткого управления недетерминированными объектами изложены в [6]. Процесс построения системы нечеткого вывода для данной задачи заключается в формализации характеристик уязвимости в контексте лингвистических переменных, определения функций принадлежности и их параметров, составление базы правил нечеткого вывода. Наиболее популярными методами алгоритмами нечеткого вывода являются Цукамото, Сугено-Такаги нулевого или первого порядка, Мамдани [6]. В данном исследовании используется алгоритм Мамдани.

Рассмотрим информацию об X_i уязвимости как некое множество характеристик $X_i = \{x_1, x_2, ..., x_n\}$. Каждая характеристика x_i может быть представлена в виде лингвистической переменной, терм-множества и семантические правила которой задаются и варьируются в зависимости от экспертной оценки. Пусть степень воздействия X_i уязвимости по отношению к продукту P является нечетким логическим выводом $I_i = f(X_i)$ с помощью нечеткой базы знаний.

[a] yullia.tatarinova@mail.com

Табл. 1. Входной набор данных для каждой CVE

Обозначение	Наименование лингвистической переменной	Описание	Терм множества	Параметры функции принадлежности
$CVSS_{base}$	Базовая метрика	Основополагающая оценка уязвимости	Низкая (Н) Средняя (С) Высокая (В) Критическая (К)	[0 0 4] [3 5.25 7.5] [6 8 10] [9 10 10]
$CVSS_{exploit}$	Возможность эксплуатации	Метод и сложность доступа, уровень аутентификации	Н С В К	[0 0 4] [[3 5.25 7.5] [6 8 10] [9 10 10]
$CVSS_{impact}$	Воздействие на систему	Метрики конфиденциальности, целостности, доступности	Н С В К	[0 0 4] [3 5.25 7.5] [6 8 10] [9 10 10]
P_{age}	Фаза состояния ПО	Жизненный цикл выпуска ПО	End-of-life Maintenance Stable release	[0 2.25 4.5] [3 5.5 8] [6 8 10]
R	Источник уязвимости	Информация о типе первопричины, источник – описание из БД	Компонент Путь к исходному файлу Фунция Переменная	[0 0 2] [0 3 6] [5 7 9] [8 10 10]
$Trend$	Тренд CVE	Степень заинтересованности сообщества ИБ к уязвимости	Низкий Средний Высокий release	[0 0 40] [25 50 75] [60 100 100]
P_{type}	Тип продукта	Категория продука, источник - описаниие из БД	firmware drivers os level service application plugin utility	[0 10 20] [10 25 40] [30 45 60] [50 65 80] [70 85 100] [90 100 100]
$Treat$	Угроза	Действия злоумышленника, источник – БД	Dos Information leakage Injection Priv escalation Cmd exec Code exec	[0 0 1] [0 2 4] [2 4 6] [4 6 8] [6 8 10] [9 10 10]

Основной задачей является построить автоматизированную систему, которая для каждой X_i уязвимости определяет степень ее воздействия I_i относительно продукта P и производит ранжирование множества I для получения рекомендаций относительно приоритизации исправлений в системе или исходном коде.

В качестве входных данных принимается исходная база данных (БД) CVE. Далее проводится анализ каждой уязвимости на основе информации из БД, дополнительных источников информации ([7]) и извлекается набор характеристик. Процесс извлечения характеристик детально описан в [8].

Для формирования входного набора лингвистических переменных используется неполное множество сформированных характеристик уязвимости в виду избыточности и нецелесообразности их фаззификации. Например, некоторые характеристик определяются как бинарные переменные: наличие кода с исправлением уязвимости, эксплойта; Common Weakness Enumeration (CWE [9]) определяет достаточно большой объем типов и классов уязвимости (более 800) и возможный результат несопоставим с трудоемкой задачей приведения данного набора к терм-множествам. Также в данной статье не учитываются характеристики, полученные из целевого программного обеспечения или устройства ([5], [8]) в целях упрощения модели. Полный перечень используемых лингвистических переменных и их определе-

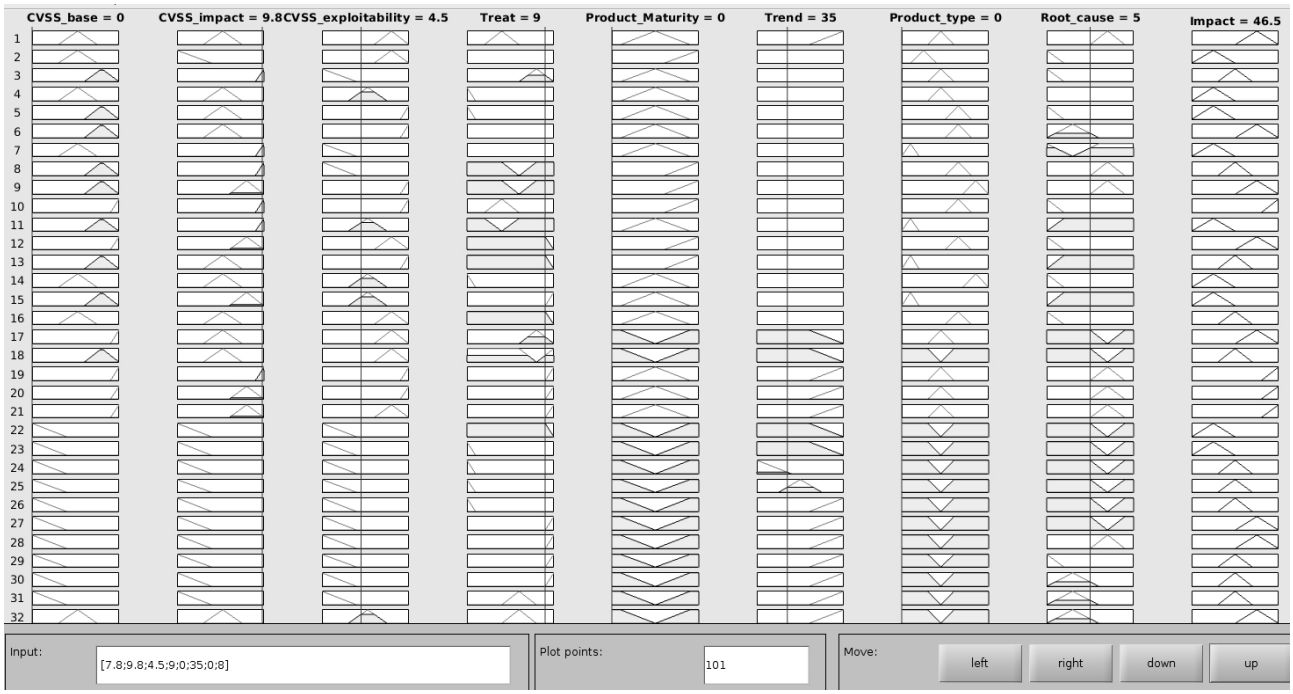

Рис. 1. Графический пример оценки CVE-2017-5130

Табл. 2. Результаты оценки нечеткой модели на произвольном наборе уязвимостей

Название	Входные данные	Оценка модели	Экспертная оценка
CVE-2018-14974	[4.8;4.25;4.5;5;5;50;50;8]	50	50
CVE-2018-0952	[7.8;4.5;9.8;5.5;9;80;50;3]	47.7	74
CVE-2018-14775	[5.5;4.5;6;0.5;0;0;50;8]	50	45
CVE-2017-2767	[9.8;9.75;9.8;9;0;0;87;7]	51	87
CVE-2017-14107	[6.5;6.9;6;0.9;7;0;87;10]	47.3	63
CVE-2017-14132	[6.5;6.9;5.8;0.9;0;0;8;10]	47.3	60
CVE-2016-7060	[4.6;6;2.25;2;0;0;50;0]	47.7	40
CVE-2016-0992	[9.8;9.8;9.75;9;5;0;78;0]	84.3	95

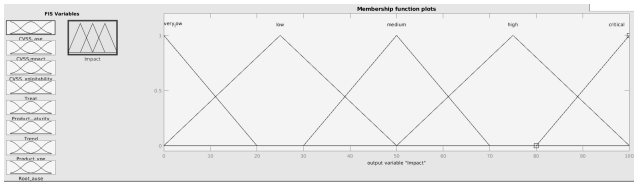

Рис. 2. Функции принадлежности входных и выходных данных.

ния показаны в таблице 1. Выходная переменная – степень воздействия уязвимости на систему.

Построение функций принадлежности и их параметров осуществляется при помощи экспертных данных эмпирическим путем. Интервалы и значения представлены в таблице 1. Активизация логического заключения в каждом из нечетких правил производится методом минимального значения. Для агрегирования значений функции принадлежности каждой из выходных переменных в заключениях нечетких правил используется метод максимального значения. Для дефаззификации используется метод центра тяжести для дискретного множества значений функции принадлежности.

3. Оценка рисков уязвимости на основе разработанной нечеткой модели

Проверка концепции производилась при помощи построения нечеткой модели в Matlab. Способ реализации был выбран из-за простоты и наглядности, однако для работы в реальных условиях в производственной среде необходимо использовать более легковесные и специализированные инструменты. Общий вид переменных показан на рисунке 2. База правил создавалась на основе случайно выбранных уязвимостей 2018 и 2017 года. Были использованы кусочно-линейные треугольные функции принадлежности. Для эксперимента количество правил равно 45. Результаты оценки работы нечеткой модели показаны в таблице 2.

Необходимо отметить, что полученный результат вывода нечеткой модели крайне отличается с экспертной оценкой на таком же наборе входных данных. Проанализировав полученные результаты на основе базы правил, можно сделать вывод, что для данного количества лингвистических переменных

и их соответствующего количества терм-множеств размер базы правил очень мал.

4. Анализ смежных работ

Использование систем нечеткого вывода для оценки рисков информационной безопасности было представлено в [10], [11], [12].

В работе [11] авторы описывают общие положение оценки рисков ИБ с использованием теории нечеткого множества. Лингвистические переменные характеризуют общие параметры, которые зачастую используются при оценке рисков: вероятность угрозы, активы и соотношение уязвимости в активах к угрозам.

Автор в [10] описывает нечеткую модель оценки риска для всей организации и берет в качестве исходных характеристик нечеткие переменные, описывающие факторы риска (организационные, технические уровни защиты информации, ценность и объем информационных ресурсов). В [10] не берутся во внимание конкретные уязвимости и их взаимодействие на целевой системе. При этом, автор определяет процесс преобразования нечеткой модели в нейро-нечеткую сеть.

В отличии от вышеперечисленных работ, данная модель предназначена для работы с техническими деталями уязвимости и конечной вычислительной системы. Более того, основная сфера применения данной модели определяется организациями, которые занимаются разработкой ПО, используют компоненты сторонних производителей и ответственны за исправление, обновление и мониторинг ошибок в своих продуктах, приложениях и системах.

5. Выводы

Полученная модель позволяет эффективно проводить оценку воздействия уязвимости на систему и определять приоритеты для своевременного устранения ошибок. Для данного набора лингвистических переменных в определенными терм-множествами в таблице 1 количество полного набора правил составляет 82944, а с учетом неполноты данных – 250880. Создание такого количества правил вручную является ресурсоемкой задачей. Более того, суммарное количество опубликованных CVE на момент написания данной статьи составляет 113522. Более того, значительная часть правил из полного набора может оказаться избыточной.

Решить подобного рода проблему можно с помощью построения адаптивной системы с использованием нечетких нейронных продукционных сетей типа Anfis [6].

В дальнейшем представленная модель будет расширена за счет добавления характеристик конкретной используемой вычислительной системы (полный набор описан в [5]). Полная структура нечеткой продукционной модели содержит несколько баз правил с соответствующими наборами лингвистических переменных. В данной работе для упрощения была представлена модель без расширений.

Перечень использованных источников

1. The CORAS Method. — `http://coras.sourceforge.net/`.
2. Humphreys Edward. Implementing the ISO/IEC 27001 information security management system standard. — Artech House, Inc., 2007.
3. Introducing octave allegro: Improving the information security risk assessment process : Rep. / Carnegie-Mellon Univ Pittsburgh PA Software Engineering Inst ; Executor: Richard A Caralli, James F Stevens, Lisa R Young, William R Wilson : 2007.
4. MITRE. Common vulnerabilities and exposures. — `https://cve.mitre.org/`.
5. Tatarinova Yuliia. AVIA: Automatic Vulnerability Impact Assessment on the Target System. — 2018. — 08. — P. 364–368.
6. Хижняков Ю. Алгоритмы нечеткого, нейронного и нечетконейронного управления в системах реального времени. — 2013.
7. Google Trends. — `https://trends.google.com/trends/?geo=US`.
8. Tatarinova Yuliia Sinelnikova Olga. Extended Vulnerability Feature Extraction Based on Public Resources. — (in press).
9. Common Weakness Enumeration. — `https://cwe.mitre.org/`.
10. Glushenko Sergey A. An adaptive neuro-fuzzy inference system for assessment of risks to an organization's information security // Бизнес-информатика. — 2017. — № 1 (39).
11. ASSESSING INFORMATION SECURITY RISK WITH THE FUZZY SET THEORY. / MURATKHAN RAIKHAN, KHABDOLDA BOLAT, ZHUMABEKOV MEIRAM, OMAROVA ALTYNAY // Journal of Theoretical & Applied Information Technology. — 2018. — Vol. 96, no. 11.
12. Сибикина Ирина Вячеславовна. Анализ рисков информационной безопасности с использованием системы нечеткого вывода // Научный вестник Новосибирского государственного технического университета. — 2016. — № 4. — С. 121–134.